MKTEMP

Temp file name easy to guess. Vulnerable to TOCTOU. (Function should not be used)

Sean Barnum, Cigital, Inc. [vita[1]]

Copyright © 2007 Cigital, Inc.

2007-04-02

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 6947 bytes

Attack Category	• Path spoofing or confusion problem
Vulnerability Category	• Temporary file creation problem • Indeterminate File/Path • TOCTOU - Time of Check, Time of Use
Software Context	• File Creation
Location	
Description	The mktemp(char *template) creates a unique temporary file using the input template. The last six characters of the template must be XXXXXX; these are replaced with a string that will make the filename unique. THIS FUNCTION SHOULD NOT BE USED. Some implementations replace the XXXXXX combination with the current process ID followed by a single letter. With only 26 possible values, it is relatively easy for an attacker to guess the filename and access the contents. It is also possible for a race condition to exist between testing whether the name exists and opening the file. mktemp() is vulnerable to TOCTOU attacks. A call to mktemp() should be unilaterally flagged. If this call must be used and if a "check present" is done, then a race condition is possible. This function creates a file; therefore there is a vulnerability (based on the above description) that the file name can be "guessed".
APIs	**Function Name** / **Comments** _mktemp / use _tmktemp / use _wmktemp / use mktemp / use
Method of Attack	The key issue with respect to TOCTOU vulnerabilities is that programs make assumptions

1. http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

	about atomicity of actions. It is assumed that checking the state or identity of a targeted resource followed by an action on that resource is all one action. In reality, there is a period of time between the check and the use that allows either an attacker to intentionally or another interleaved process or thread to unintentionally change the state of the targeted resource and yield unexpected and undesired results The mktemp() call is a use-category call, which when preceded by a check-category call can be indicative of a TOCTOU vulnerability. See description. File name can be relatively easily guessed.
Exception Criteria	

Solutions

Solution Applicability	Solution Description	Solution Efficacy
Universal application	Do not ever use mktemp. The mkstemp function avoids the race condition because the file is opened with the O_EXCL flag, guaranteeing that when mkstemp returns successfully the program is the only user.	Effective
Generally applicable.	The most basic advice for TOCTOU vulnerabilities is to not perform a check before the use. This does not resolve the underlying issue of the execution of a function on a resource whose state and identity cannot be assured, but it does help	Does not resolve the underlying vulnerability but limits the false sense of security given by the check.

		to limit the false sense of security given by the check.	
	Generally applicable.	Limit the interleaving of operations on files from multiple processes.	Does not eliminate the underlying vulnerability but can help make it more difficult to exploit.
	Generally applicable.	Limit the spread of time (cycles) between the check and use of a resource.	Does not eliminate the underlying vulnerability but can help make it more difficult to exploit.
	Generally applicable.	Recheck the resource after the use call to verify that the action was taken appropriately.	Effective in some cases.

Signature Details	char *mktemp(char *template); int mkstemp(char *template);
Examples of Incorrect Code	
Examples of Corrected Code	
Source References	• Viega, John & McGraw, Gary. *Building Secure Software: How to Avoid Security Problems the Right Way*. Boston, MA: Addison-Wesley Professional, 2001, ISBN: 020172152X, ch. 9. • man page for mktemp() • Microsoft Developer Network Library (MSDN)
Recommended Resource	
Discriminant Set	**Operating Systems**: • UNIX • Windows **Languages**: • C • C++

Cigital, Inc. Copyright

Copyright © Cigital, Inc. 2005-2007. Cigital retains copyrights to this material.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

For information regarding external or commercial use of copyrighted materials owned by Cigital, including information about “Fair Use,” contact Cigital at copyright@cigital.com[1].

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. mailto:copyright@cigital.com